Idea

This program is for the express edition Pokemon Go game. You can travel around the “map”; select your first Pokemon buddy; meet and catch the Pokemon; and also, you can save the game and load it.

Inputs/Outputs

[Input] The choice for the following question

What do you want to do?
1.Start your new adventure!
2.Continue your game!
3.Quit!

Which Pokemon you want to choose to be your first buddy?
1.squirtle, 2. bulbasaur, 3. charmander
Notes: Enter the number!

1.Go on adventure!
2.Check your Pokemons!
3.Save your game!
4.Quit

Where you want to go? (W for up, S for down, D for right, A for left)

You want to capture or not? (Y/N)

[Output] The “map” and where the adventurers are
[Output] The Pokemon adventurers have
[Output] record.txt

Variables

Class game_adventurer() for the users, has the three attributes:

1. The position for users as array
2. The Pokemon users have already had as list
3. The map with the position the users are as array

Class pokemon() for the Pokemons the adventurers meet, has the three attributes:

1. The name as string

2. The value for meeting (use for the calculate the probability for meeting the Pokemon) as int
3. The value for catching (use for the calculate the probability for catching the Pokemon) as int

The record.txt to save the record of the game as file

List of Task

1. Travel around the "map" and show the map with the position the users
   ·We get the class game_adventurer() first and also with its three attributes
   I. Add the method show(self) to show the map
      a. Position is two-coordinate array, can points the "O" out on the map;
      b. Using the array and the for loop to show the map, "_" is the unknown place and "O" is the place where the users are.
   II. Add the method move(self,n) to display "traveling around the map"
      a. Give the option (Where you want to go? (W for up, S for down, D for right, A for left)
      b. According to the users' choice to change the two-coordinate array (position)
      c. And add the try & except to avoid the IndexError, which means that the users may be go out of the map.
2. Show the Pokemons the users have already had
   Still use the class game_adventurer()
   I. Pokemon is a list
      a. Every time the users catch one Pokemon and we will append its name with the list
      b. When the users want to check, just print the game_adventurer().pokemon
3. Meet the Pokemon and catch it
   Use the class pokemon() and with its three attribute
   I. We randomly choose a name and the value for meeting and catching (0-10)
   II. Add the method rate_meet(self) to calculate the probability for meeting the Pokemon
      a. If the value for meeting is bigger than 8, meaning that the users will not meet the Pokemon
      b. Else, meaning that they can meet the Pokemon
   III. Add the method rate_capture(self) to calculate the probability for catching the Pokemon
      a. If the value for catching is bigger than 6, meaning that the users will not catch the Pokemon this time
      b. Else, meaning that they can catch the Pokemon
   IV. When every time users try to move, they will have the chance to meet and catch the Pokemon

4. Save the game and load it
   I. Add the function choose_3(user) for saving the game (user= game_adventurer() is the class)
   For my game, the position for the users and the Pokemon that users have should be saved
      a. Use the “with open (“record.txt”, “w”) as out” to write, the first two line is for the two-coordinate array (Position)
      b. Then use the for loop to save the pokemon list, every single pokemon write the one single line
   II. Add the function read_record(user) for loading the game (user= game_adventurer() is the class)
      a. Use the “with open (“record.txt”, “r”) as out” to read, the first assign to the user(class) position
      b. Then use the for loop to read the pokemon list, appending them together as a list
      c. Use the try & except to avoid IOError, which means that avoiding the “record.txt” is not existed.
5. Show the main menu and the sub menu
   I. Add the function menu () to save the choice for the following question as integer,
   What do you want to do?
   1.Start your new adventure!
   2.Continue your game!
   3.Quit!
   II. Add the function sub_menu () to save the choice for the following question as integer,
   1.Go on adventure!
   2.Check your Pokemons!
   3.Save your game!
   4.Quit
6. Using two functions to combine the sub_menu and menu with the choice it can choose. (using while loop and if & else)

Requirement:

[Requirement one]:

I. My game has the “main character” and user can control the character for using “W”,” A”,” S”,” D”.
II. Using “W”,” A”,” S”,” D” can change the position
III. The user has three attributes (position, pokemon, map)

[Requirement two]:

I. My game has the board and also has 9×9 possible positions

[Requirement three]:

I. My game has the position on the board, "O" is the position where the user is
II. Every movement for the users, they can meet the pokemon and also can catch it

[Requirement Four]:

My game has four topics from the last third of the course

I. **File I/O** I use this topic to save and load the game state (In functions choose_3(user) and read_record(user))
II. **Error handling (try/except)** I use this topic to avoid IOError, which means that avoiding the "record.txt" is not existed; to avoid IndexError, which means that the users may be go out of the map; to avoid ValueError, which means that the users do not enter the number 1-3 to select the choice, they may enter the characters.
III. **Classes/objects** I use this topice to create two class, one for game_adventurer(), the other one is for the pokemon （）
IV. **NumPy** I use this topice to display the map and show where the user is.